AE 23: Logistic regression

Nico Robin

2021-11-19

```
library(tidyverse)
library(tidymodels)
library(boot)
```

Bulletin

- Due today
  - Project report (in GitHub repo)
- Upcoming
  - Peer review in lab tomorrow
  - homework 05 released today
  - complete survey on community

Learning goals

By the end of today, you will...

- understand logistic regression as a linear model of binary outcomes
- be able to fit logistic regression in R

To illustrate logistic regression, we will build a spam filter from email data. Today's data consists of 4601 emails that are classified as `spam` or `non-spam`. The data was collected at Hewlett-Packard labs and contains 58 variables. The first 48 variables are specific keywords and each observation is the percentage of appearance (frequency) of that word in the message. Click here to read more.

- `type` = 1 is spam
- `type` = 0 is non-spam

```
spam = read_csv("data/spam.csv")
glimpse(spam)
```

```
## Rows: 4,601
## Columns: 58
## $ make              <dbl> 0.00, 0.21, 0.06, 0.00, 0.00, 0.00, 0.00, 0.00, 0.15~
## $ address           <dbl> 0.64, 0.28, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00~
## $ all               <dbl> 0.64, 0.50, 0.71, 0.00, 0.00, 0.00, 0.00, 0.00, 0.46~
## $ num3d             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ our               <dbl> 0.32, 0.14, 1.23, 0.63, 0.63, 1.85, 1.92, 1.88, 0.61~
## $ over              <dbl> 0.00, 0.28, 0.19, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00~
## $ remove            <dbl> 0.00, 0.21, 0.19, 0.31, 0.31, 0.00, 0.00, 0.00, 0.30~
## $ internet          <dbl> 0.00, 0.07, 0.12, 0.63, 0.63, 1.85, 0.00, 1.88, 0.00~
## $ order             <dbl> 0.00, 0.00, 0.64, 0.31, 0.31, 0.00, 0.00, 0.00, 0.92~
## $ mail              <dbl> 0.00, 0.94, 0.25, 0.63, 0.63, 0.00, 0.64, 0.00, 0.76~
```

```
## $ receive           <dbl> 0.00, 0.21, 0.38, 0.31, 0.31, 0.00, 0.96, 0.00, 0.76~
## $ will              <dbl> 0.64, 0.79, 0.45, 0.31, 0.31, 0.00, 1.28, 0.00, 0.92~
## $ people            <dbl> 0.00, 0.65, 0.12, 0.31, 0.31, 0.00, 0.00, 0.00, 0.00~
## $ report            <dbl> 0.00, 0.21, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00~
## $ addresses         <dbl> 0.00, 0.14, 1.75, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00~
## $ free              <dbl> 0.32, 0.14, 0.06, 0.31, 0.31, 0.00, 0.96, 0.00, 0.00~
## $ business          <dbl> 0.00, 0.07, 0.06, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00~
## $ email             <dbl> 1.29, 0.28, 1.03, 0.00, 0.00, 0.00, 0.32, 0.00, 0.15~
## $ you               <dbl> 1.93, 3.47, 1.36, 3.18, 3.18, 0.00, 3.85, 0.00, 1.23~
## $ credit            <dbl> 0.00, 0.00, 0.32, 0.00, 0.00, 0.00, 0.00, 0.00, 3.53~
## $ your              <dbl> 0.96, 1.59, 0.51, 0.31, 0.31, 0.00, 0.64, 0.00, 2.00~
## $ font              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ num000            <dbl> 0.00, 0.43, 1.16, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00~
## $ money             <dbl> 0.00, 0.43, 0.06, 0.00, 0.00, 0.00, 0.00, 0.00, 0.15~
## $ hp                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ hpl               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ george            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ num650            <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00~
## $ lab               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ labs              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ telnet            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ num857            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ data              <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.15~
## $ num415            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ num85             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ technology        <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00~
## $ num1999           <dbl> 0.00, 0.07, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00~
## $ parts             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ pm                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ direct            <dbl> 0.00, 0.00, 0.06, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00~
## $ cs                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ meeting           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ original          <dbl> 0.00, 0.00, 0.12, 0.00, 0.00, 0.00, 0.00, 0.00, 0.30~
## $ project           <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00~
## $ re                <dbl> 0.00, 0.00, 0.06, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00~
## $ edu               <dbl> 0.00, 0.00, 0.06, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00~
## $ table             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ conference        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ charSemicolon     <dbl> 0.000, 0.000, 0.010, 0.000, 0.000, 0.000, 0.000, 0.0~
## $ charRoundbracket  <dbl> 0.000, 0.132, 0.143, 0.137, 0.135, 0.223, 0.054, 0.2~
## $ charSquarebracket <dbl> 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.0~
## $ charExclamation   <dbl> 0.778, 0.372, 0.276, 0.137, 0.135, 0.000, 0.164, 0.0~
## $ charDollar        <dbl> 0.000, 0.180, 0.184, 0.000, 0.000, 0.000, 0.054, 0.0~
## $ charHash          <dbl> 0.000, 0.048, 0.010, 0.000, 0.000, 0.000, 0.000, 0.0~
## $ capitalAve        <dbl> 3.756, 5.114, 9.821, 3.537, 3.537, 3.000, 1.671, 2.4~
## $ capitalLong       <dbl> 61, 101, 485, 40, 40, 15, 4, 11, 445, 43, 6, 11, 61,~
## $ capitalTotal      <dbl> 278, 1028, 2259, 191, 191, 54, 112, 49, 1257, 749, 2~
## $ type              <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
```

The basic logic of our model is that the frequency of certain words can help us determine whether or not an email is spam.

For example, these emails came from George's inbox. If the word "george" is not present in the message and the dollar symbol (`charDollar`) is, you might expect the email to be spam.

Using this data, we want to build a model that **predicts** whether a new email is spam or not. How do we build a model that can do this?

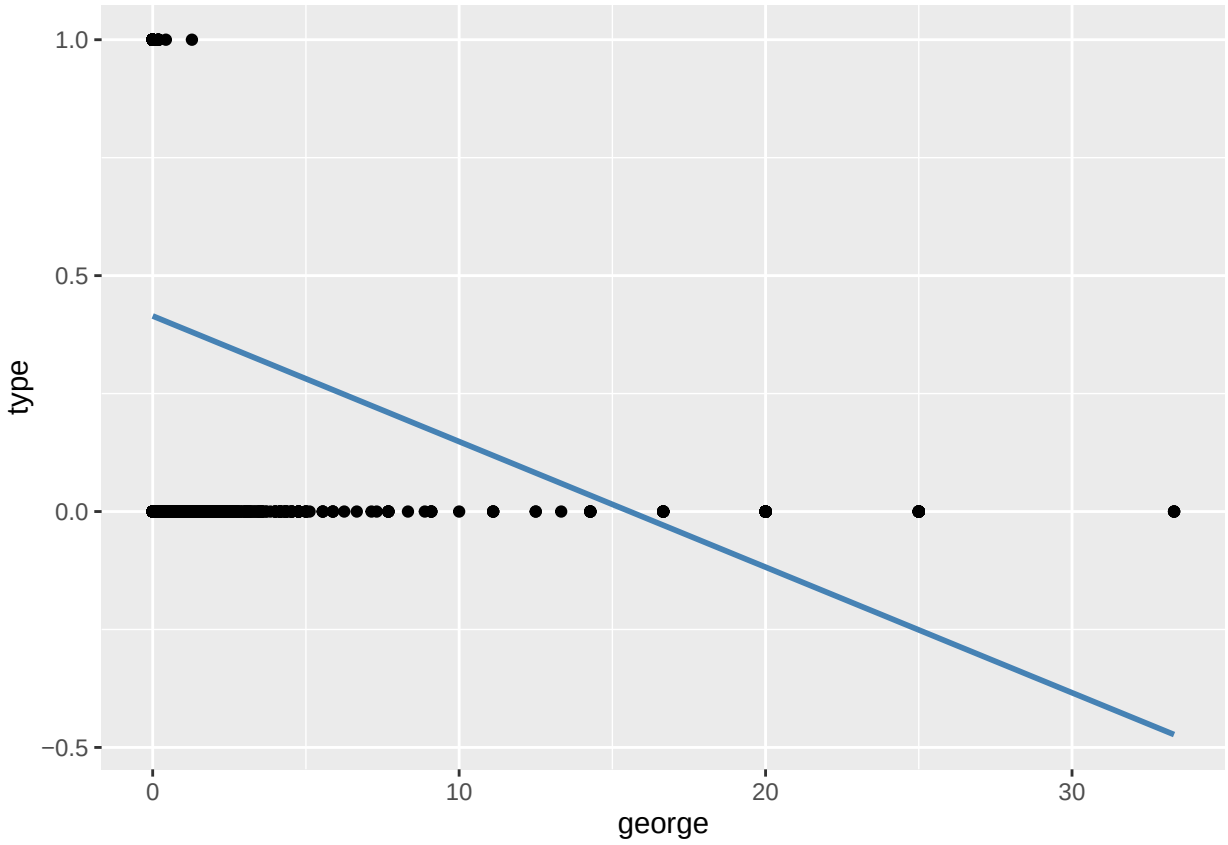
Exercise 1

Start by examining 1 predictor.

- Visualize a linear model where the outcome is `type` (spam or not) and `george` is the predictor.
- Discuss your visualization with your neighbor. Is this a good model? Why or why not?

```
spam %>%
  ggplot(aes(x = george, y = type)) +
  geom_point() +
  geom_smooth(method = "lm", color = 'steelblue', se = F)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



The regression predicts outcome values that extend over the entire real line, but our outcome is simply 0 (non-spam) or 1 (spam).

Logistic regression

How do you build a model to fit a binary outcome?

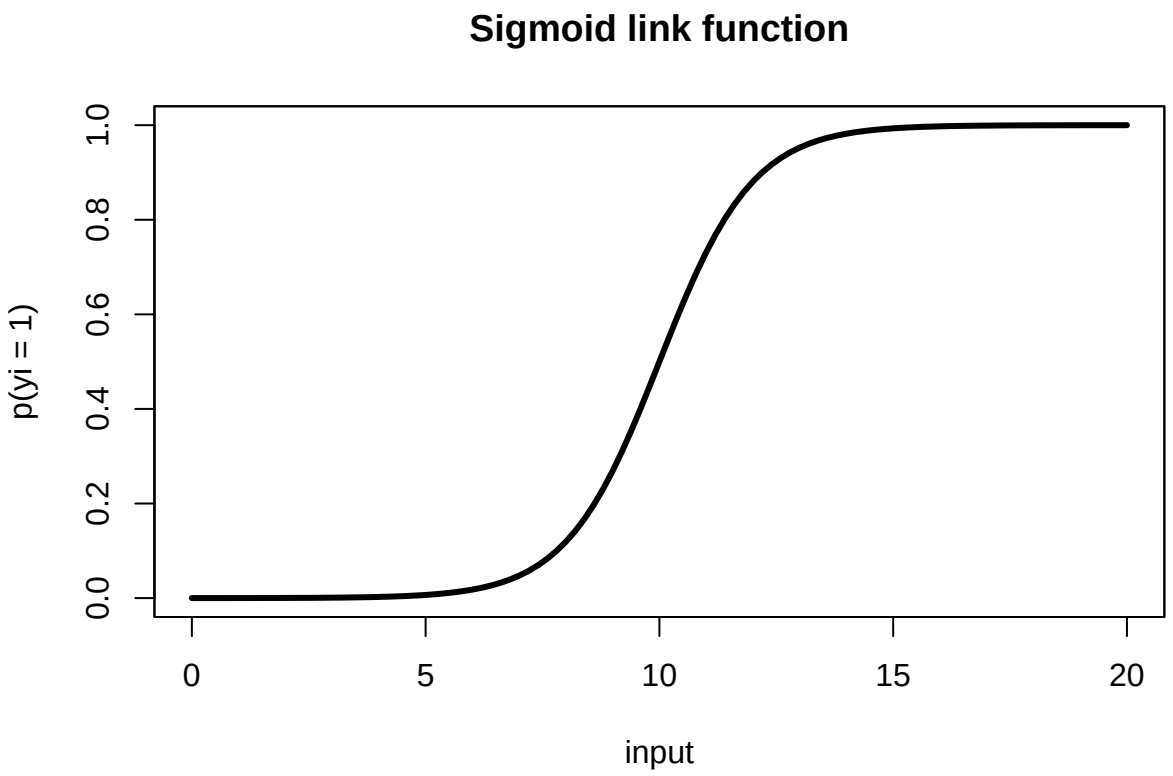
Linear logistic regression (also simply called "logistic regression") takes in a number of predictors and outputs the probability of a "success" (an outcome of 1) in a binary outcome variable. The probability is related to

the predictors via a **sigmoid link function**,

$$p(y_i = 1) = \frac{1}{1 + \exp\{-\sum \beta_i x_i\}},$$

whose output is in $(0, 1)$ (a probability). In this modeling scheme, one typically finds $\hat{\beta}$ by maximizing the **likelihood function**, (another objective function, different than our previous "least squares" objective).

```
sigmoid = function(x) 1 / (1 + exp(-x + 10))
plot.function(sigmoid, from = 0, to = 20, n = 101, ylab="p(yi = 1)", xlab="input", main="Sigmoid link fu
box()
```



To proceed with building our email classifier, we will, as usual, use our data (outcome y_i and predictor x_i pairs), to estimate β (find $\hat{\beta}$) and obtain the model:

$$p(y_i = 1) = \frac{1}{1 + \exp\{-\sum \hat{\beta}_i x_i\}},$$

Example

Let's build a model centered around just two predictor variables.

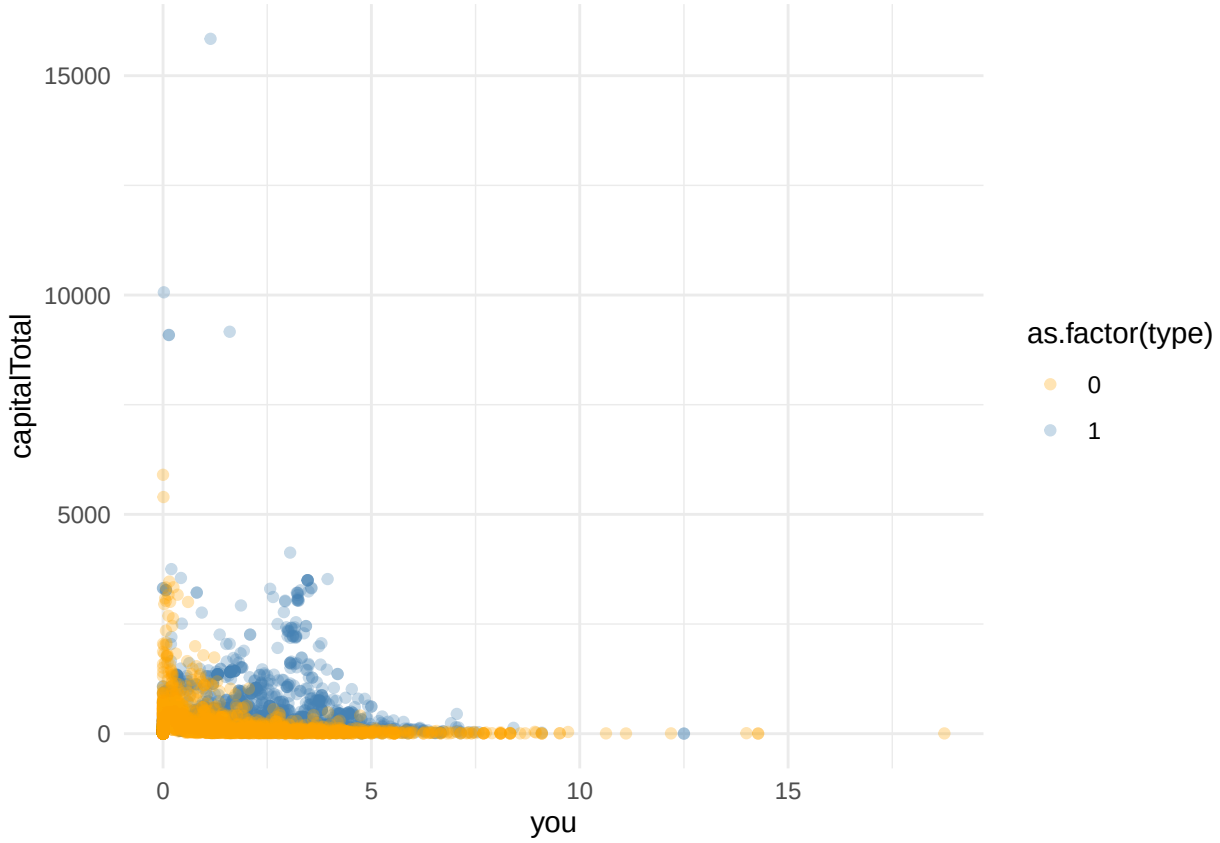
The first will be the word `you` and the second will be `capitalTotal` (the total number of capital letters in the message).

Exercise 2

Create a visualization with `you` on the x-axis and `capitalTotal` on the y-axis. Color data points by whether or not they are spam.

```
spam %>%
  ggplot(aes(x = you, y = capitalTotal, color = as.factor(type))) +
```

```
  geom_point(alpha = 0.3) +
  scale_colour_manual(values = c("orange", "steelblue")) +
  theme_minimal()
```




```
fit_1 = logistic_reg() %>%
  set_engine("glm") %>%
  fit(as.factor(type) ~ you + capitalTotal, data = spam, family = "binomial")

fit_1 %>%
  tidy()
```

```
## # A tibble: 3 x 5
##   term         estimate std.error statistic   p.value
##   <chr>           <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)  -1.50     0.0554       -27.1 2.97e-162
## 2 you           0.361    0.0198        18.3 1.84e- 74
## 3 capitalTotal  0.00173  0.000104      16.6 5.66e- 62
```

Exercise 3

- What is different in the code above from previous linear models we fit?

`logistic_reg()`, engine `glm`, family `binomial` (really this family binomial bit is optional)

Exercise 4

- What is the probability the email is spam if the frequency of `you` is 5% in the email and there are 2500 capital letters. Use the model equation above.

- What is the log-odds? (Recall from the prep that log-odds $= \frac{p}{1-p}$). Use the code below to check your work.

```
newdata = data.frame(you = 5, capitalTotal = 2500)

# code here
linearPart =  -1.502575519 + (.361040108 * 5) + (0.001732204 * 2500)
p = 1 / (1 + exp(-1*linearPart))
p
```

```
## [1] 0.9903694
```

```
manualLogOdds = log(p / (1 - p))
manualLogOdds
```

```
## [1] 4.633135
```

```
# check work
checkLogOdds = predict(fit_1$fit, newdata)
checkLogOdds
```

```
##        1
## 4.633134
```

```
checkP = inv.logit(checkLogOdds)
checkP
```

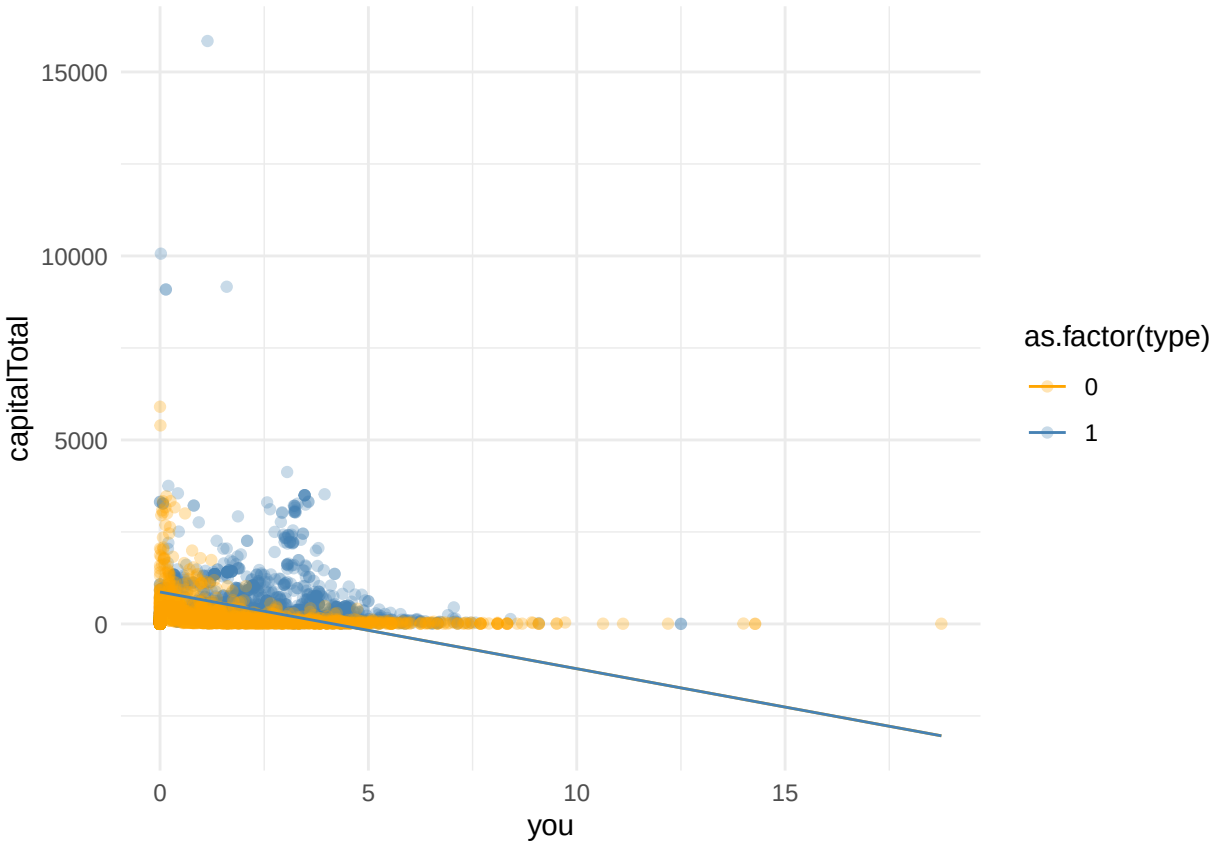
```
##         1
## 0.9903694
```

Visualize logistic regression

```
beta = fit_1$fit$coefficients
hyperplane = function(x){
    decisionBoundary = 0.5
    c = logit(decisionBoundary)
    const = c - beta[1]
    return((-beta[2]*x + const) / beta[3])
}

spam %>%
  ggplot(aes(x = you, y = capitalTotal, color = as.factor(type))) +
  geom_point(alpha = 0.3) +
  geom_function(fun = hyperplane) +
  scale_colour_manual(values = c("orange", "steelblue")) +
  theme_minimal()
```

```
## Warning: Multiple drawing groups in `geom_function()`. Did you use the correct
## `group`, `colour`, or `fill` aesthetics?
```



- Just because there's greater than 50% probability an email is spam doesn't mean we have to label it as such. We can adjust our **threshold** or **critical probability**, a.k.a. **decision boundary** to be more or less sensitive to spam emails.

In other words we get to select a number p^* such that

if $p > p^*$, then label the email as spam.

Exercise 5

- What would you set your decision boundary to and why?
- Change `decisionBoundary` in the code above to 0.01 and 0.999999. Do the results surprise you? Why or why not?
- lower boundary means that we label more emails as spam, high boundary means fewer emails as spam. We can adjust the boundary depending on how much we value receiving important emails vs how much we dislike spam.
- 0 means all emails are spam, 1 means no emails are spam. Note you cannot set decision boundary to 0 or 1 because of logit function (would evaluate to inf or negative inf)

Classify a new email

```
email = readLines("data/test-email.txt")
email
```

```
## [1] "You Have Been Selected To Win A Free Trip To Disney World! "
## [2] ""
## [3] "YOU HAVE 30 SECONDS TO CLICK HERE TO CLAIM YOUR REWARD!"
```

```
## [4] ""
## [5] "WHAT ARE YOU WAITING FOR? ACT NOW!"
## [6] ""
## [7] "SINCERELY,"
## [8] ""
## [9] "WALT DISNEY"
```

```
totalWord = sum(str_count(email, " "))
totalYou = sum(str_count(tolower(email), "you"))
capitalTotal = sum(str_count(email, "[A-Z]"))

youFreq = 100 * totalYou / totalWord
newemail = data.frame(you = youFreq, capitalTotal = capitalTotal)

logOdds = predict(fit_1$fit, newemail)
logOdds
```

```
##        1
## 3.648776
```

```
inv.logit(logOdds)
```

```
##         1
## 0.9746371
```

Exercise 6

- Does the code above count the correct number of "you"? Why or why not?
- Do you believe the predicted odds of the email being spam? Why or why not?
- What is the **probability** the test email is spam?

solutions

- no it also counts "your"
- Yes, the email seems to be spam and the odds are $e^{\text{log-odds}} = 38:1$
- the probability is 0.97 or 97% probability.